JAVA Run Time Environment

JVM, JRE, JDK

- **Java Virtual machine (JVM)** is the virtual machine that runs the Java bytecodes. You get this bytecode by compiling the .java files into .class files. .class files contain the bytecodes understood by the JVM.

- Provides a runtime environment in which Java bytecode can be executed.

The JVM performs the following main tasks:
- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM, JRE, JDK

- **The Java Runtime Environment (JRE)** is a software package which bundles the libraries (jars) and the Java Virtual Machine, and other components to run applications written in the Java. JVM is just a part of JRE distributions.

- The Java Runtime Environment is a set of software tools which are used for developing Java applications.
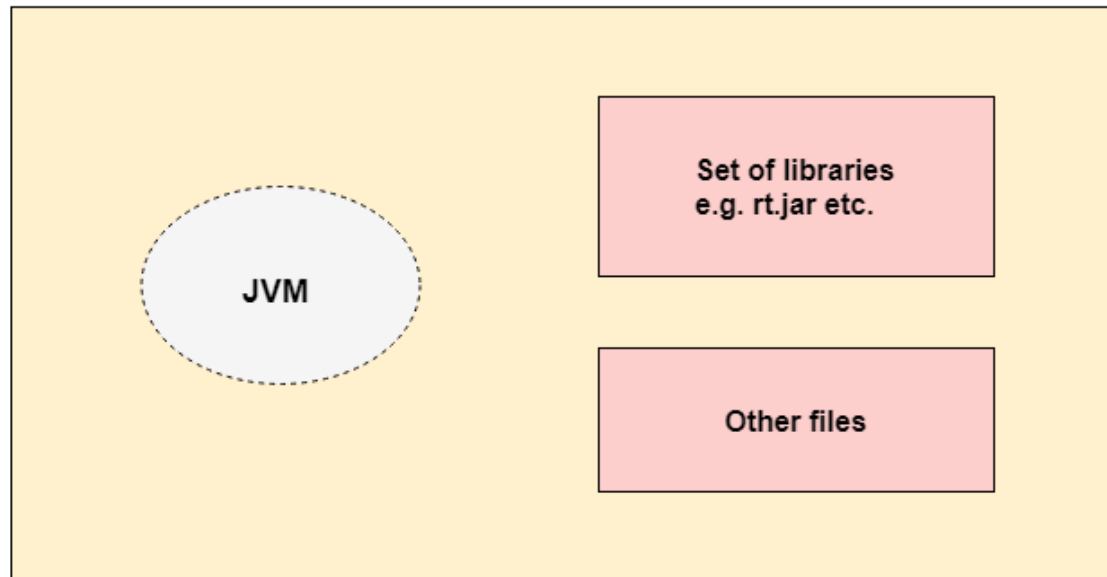
JVM, JRE, JDK

- **JDK** is an acronym for **Java Development Kit**. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets.

- The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.

Data Types

There are two data types available in Java −

Primitive Data Types

The primitive data types include boolean, char, byte, short, int, long, float and double.
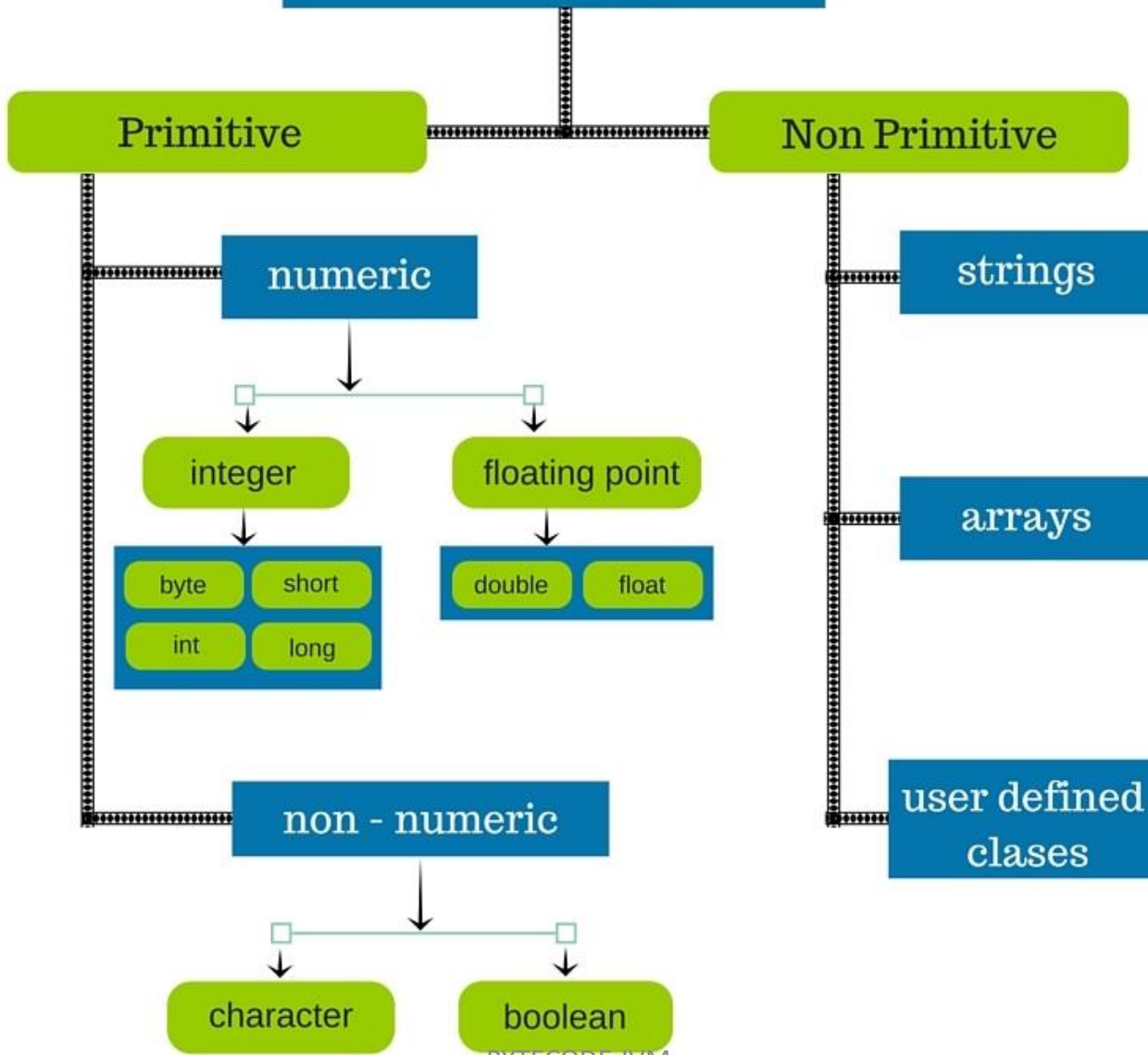
Reference/Object Data Types

The non-primitive data types include Classes, Interfaces, and Arrays.

Specify the type of data and the length of the data item in bytes

- int, short, long
- float, double
- boolean
- char

Data Types

Reserved Word	Data Type	Size	Range of Values
byte	Byte Length Integer	1 bytes	-2^8 to 2^7-1
short	Short Integer	2 bytes	-2^{16} to $2^{16}-1$
int	Integer	4 bytes	-2^{32} to $2^{31}-1$
long	Long Integer	8 bytes	-2^{64} to $2^{63}-1$
float	Single Precision	4 bytes	-2^{32} to $2^{31}-1$
double	Real number with double	8 bytes	-2^{64} to $2^{62}-1$
char	Character (16 bit unicode)	2 bytes	0 to $216-1$
boolean	Has value true or false	A boolean value	true or false

Keywords

- Java keywords are also known as reserved words.

- Keywords are particular words which acts as a key to a code

- These are predefined words by Java so it cannot be used as a variable or object name.

Keywords

abstarct	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Operators

Java provides a rich set of operators to manipulate variables.

We can divide all the Java operators into the following groups

- Arithmetic Operators

- Relational Operators

- Bitwise Operators

- Logical Operators

- Assignment Operators

- Unary Operators

- Ternary Operators

- Misc Operators

Operators

Operators	Associativity	Type
++ --	Right to left	Unary postfix
++ -- + - ! (type)	Right to left	Unary prefix
/ * %	Left to right	Multiplicative
+ -	Left to right	Additive
< <= > >=	Left to right	Relational
== !==	Left to right	Equality
&	Left to right	Boolean Logical AND
^	Left to right	Boolean Logical Exclusive OR
|	Left to right	Boolean Logical Inclusive OR
&&	Left to right	Conditional AND
||	Left to right	Conditional OR
?:	Right to left	Conditional
= += -= *= /= %=	Right to left	Assignment

Structure of Java Program

Documentation Section
Package Statement
Import Statement
Interface Statement
Class Definition

Main Method Class
{
    //Main method defintion
}

Structure of Java

Documentation Section

You can write a comment in this section.

Package statement

- You can create a package with any name. A package is a group of classes that are defined by a name.

Import statements

- This line indicates that if you want to use a class of another package, then you can do this by importing it directly into your program.

Interface statement

- Interfaces are like a class that includes a group of method declarations. It's an optional section and can be used when programmers want to implement multiple inheritances within a program.

Structure of Java

Class Definition

• A Java program may contain several class definitions. Classes are the ma

Main Method Class

• Every Java stand-alone program requires the main method as the starting point of the program. This is an essential part of a Java program.

Structure of Java

```
      package com.mm;                    ──────── package declaration

      import java.util.Date;             ──────── import statements

5.    /**
      * @author tutorialkart.com         ──────── comments
      */
      public class ProgramStructure {    ──────── class name

10.          int repetetions = 3;         ──────── global variable

      public static void main(String[] args){
              ProgramStructure programStructure = new ProgramStructure();      main
              programStructure.printMessage("Hello World. I started learning Java.");   method
15.   }

      public void printMessage(String message){
              Date date = new Date();        variable local to the method
              for(int index=0;index < repetetions;index++){                   method
20.                   System.out.println(message+" From "+date.toGMTString());
              }        variable local to the for loop
      }
      }
```

Example Program

```
class HelloWorld
{
  // Your program begins with a call to main().
 // Prints "Hello, World" to the terminal window.
 public static void main(String args[])
 {
    System.out.println("Hello, World");
 }
}
```

Example Program

- **Class definition:** This line uses the keyword **class** to declare that a new class is being defined.class

  HelloWorld

- **HelloWorld** is an identifier that is the name of the class. The entire class definition, including all of its members, will be between the opening curly brace **{** and the closing curly brace **}** .

- **main method:** In Java programming language, every application must contain a mainmethod whose signature is:

  public static void main(String[] args)

  public: So that JVM can execute the method from anywhere.

  static: Main method is to be called without object. The modifiers public and static can be written in either order.

  void: The main method doesn't return anything.

  main(): Name configured in the JVM.

  String[]: The main method accepts a single argument: an array of elements of type String.

Example Program

System.out.println("Hello, World");

- This line outputs the string "Hello, World" followed by a new line on the screen.

- Output is actually accomplished by the built-in println( ) method.

- **System** is a predefined class that provides access to the system, and **out** is the variable of type output stream that is connected to the console.